Identity Assurance

(Scott Rea – Dartmouth College)

IdM Workshop, Brisbane Australia, August 19, 2008

DARTMOUTH COLLEGE

Contents

DARTMOUTH COLLEGE

Authentication & Identity Assurance

Identity Assurance

- What is Identity Assurance?

Identity Assurance is a measure of the confidence that the entity at the other end of an authentication event, is who they are claiming to be

- Identity Assurance (IA) is a foundational element for effective security.
- IA is a pre-requisite for effective identity management, and identity management a pre-requisite for robust security

Authentication

- Authentication is the reliable identification of an entity based upon the presentation of a previously assigned token
  - E.g. An account is created for Student X in the campus directory, the first time the student logs into the directory, they are asked to create a new password – this password becomes their authentication token
  - Subsequently when a student wants to access services on campus, a given service will ask which account (e.g. username for Student X) the student wants to access, then verifies who the requestor is by asking for their authentication token (in this case the password)
  - If the requestor provides the correct password for the account, then the service assumes that the requestor is in fact the owner of the account (i.e. Student X) and can then make further authorization decisions upon that basis i.e. Student X is authenticated

Authentication Strength

- Security is like a chain, it is only as strong as its weakest link
- The strength of an authentication event determines what trust or assurance can be placed in the assumption that you are in fact dealing with the claimed identity
- The strength of any Authentication event is dependent on the following:
  - The original process to bind the identity to the authentication token
  - The life cycle management and protection of the authentication token by the identity
  - The infrastructure and protocols used by a service to validate an authentication token
  - The use of multiple authentication factors to verify identity

Authentication Factors

- Three Factors of Authentication:
  - Something you know
    - e.g. password, secret, URI, graphic
  - Something you have
    - e.g. key, token, smartcard, badge
  - Something you are
    - e.g. fingerprint, iris scan, face scan, signature

Authentication Factors

- Single Factor of Authentication is most common
  - Passwords (something you know) are the most common single factor
- At least Two Factor Authentication is recommended for securing important assets
  - e.g. ATM card + PIN (have + know)
- 2 x Single Factor Authentication ≠ Two Factor Authentication
  - e.g. Password + Graphic is NOT equivalent to Smartcard + PIN (although it may be better than a single instance of One Factor Authentication)
- Without Two Factor Authentication, some secure communications may be vulnerable to disclosure
  - Especially in wireless networks

Authentication Infrastructure

- There are essentially 2 main factors that impact the assurance of an authentication event due to infrastructure
  - Is there a secure path between claimant and service for transmitting identity credentials
    - E.g. TLS for open network traversal or dedicated line owned by the service provider
    - Integrity of the operating environments at either end of the communications pipe
  - Availability of trusted and up-to-date validity status of the presented credentials
    - E.g. Access to authoritative directory for verification of password
    - CRLs or OCSP for PKI credentials

Authentication Token Life-cycle

- Another important consideration for the assurance in an authentication event, is the management of the authentication token by the holder AFTER it was issued
  - E.g. for Username/Password system
    - Was an Acceptable Use Policy agreed to by the user and if so, is it being followed?
    - Has the User shared their password? Are they changing it at the required intervals? Are they using strong password characteristics? Did they write down or leave it unprotected? Do they avoid shoulder surfers? Did they change their password after they discovered it was compromised?
  - E.g. for PKI
    - Was a Subscriber Agreement agreed to by the user and if so, is it being followed?
    - Is the user keeping their private key password protected? Did they move their private key to a public location? Are they using the credential for an unsanctioned purpose? Did they share their private key with another user? Did they revoke their certificate after they discovered it was compromised?

Identity Binding

- The strength of an authentication event is directly dependent on the original identity binding process utilized when the authentication or identity token was issued
  - How was the original identity verified?
    - What processes were used to ensure the subscriber/user is the rightful owner of their claimed identity?
    - Was trusted biometric verification used e.g. photo ID or fingerprint from a trusted authority?
    - How reliable are the sources of identity information?
    - Is there a valid reason for the entity to obtained the credential?
  - What type of credential was issued?
    - Is it resistant to tampering, counterfeit, or exploitation?
    - Does it have on-going validity assertion capability?

Authentication Strength

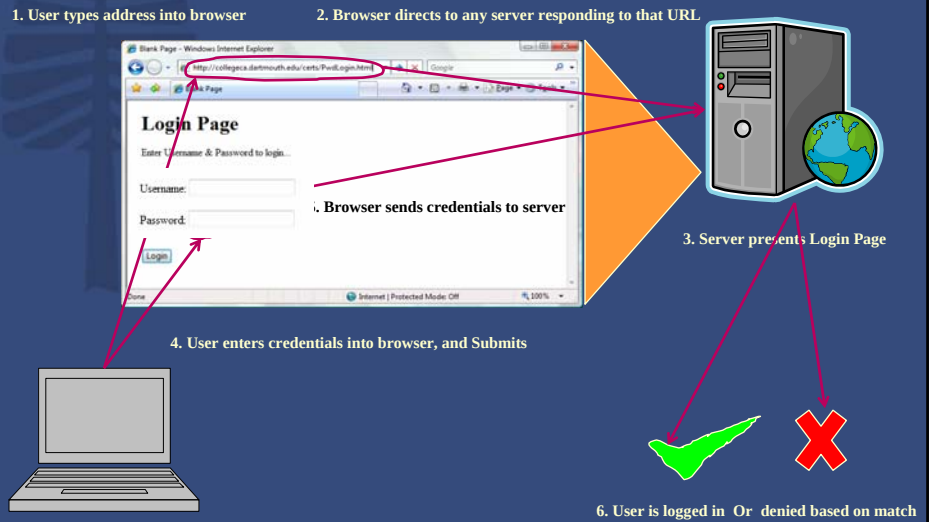
- Security is like a chain, it is only as strong as its weakest link
- The strength of an authentication event determines what trust or assurance can be placed in the assumption that you are in fact dealing with the claimed identity
- The strength of any Authentication event is dependent on the following:
  - The original process to bind the identity to the authentication token
  - The life cycle management and protection of the authentication token by the identity
  - The infrastructure and protocols used by a service to validate an authentication token
  - The use of multiple authentication factors to verify identity

Authentication Strength Comparison

- The following set of slides is a comparison of some common authentication tokens and their mechanisms as used on Dartmouth campus
  - They are presented in order of strength from Dartmouth's perspective along the Identity Assurance continuum
  - A brief discussion for each authentication token/mechanism is detailed on how it has the capability to affect the assurance of transactions based upon them

Password Authentication

Plain Password Authentication



Plain Password Authentication

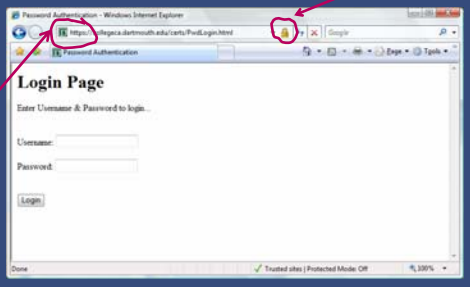
- By using an "http" URL, the user has no guarantee that they are talking to the correct server
- There are no transport protections so username and password can be intercepted and stolen in transit (wired or wireless)
- A MITM attacker simply pretends to be the server (local DNS poisoning) , asks the user for their credentials and replays the answers to the real server in real time to gain access – the user is oblivious to this attack
- If the password is saved in the browser, it can be stolen by malware or a malicious user
- A user can be socially engineered to reveal username /password to an attacker

- Passwords only provide a single authentication factor
- Passwords generally represent a poor binding between identity and credential
- The server knows everyone's password – it is a single point of failure for ALL credentials if it is compromised
- A malicious server can masquerade as any user to any other service that accepts the same username/ password (MITM)
- The server has very minimal assurance that they are talking to the original user – the password could have been shared, stolen or guessed

Server-side PKI Authentication

Adding Server-side PKI to Password Authentication

- Server-side PKI can strengthen password authentication by adding a SSL/TLS certificate to the web server and requiring users to connect over HTTPS
- If the server certificate is issued by a Root CA that is trusted by the browser, the user is connected to the web site securely and all subsequent traffic between the server and the browser is encrypted

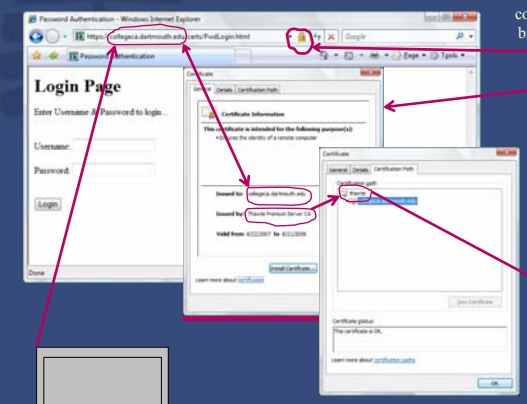


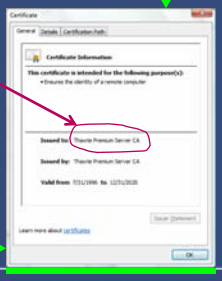
Adding Server-side PKI to Password Authentication

The name of the web server certificate matches what the user typed into the browser

The lock indicates a secure connection between browser and server

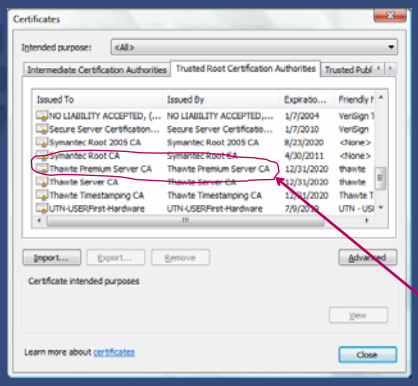
Details about the web server certificate can be viewed by clicking on the lock in the browser status bar





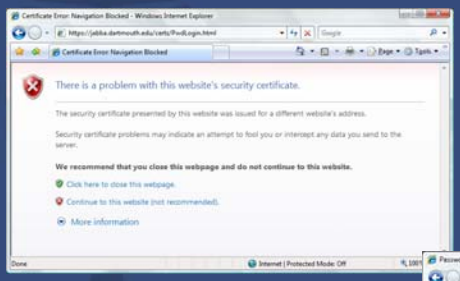
The Root CA that issued the web server certificate is trusted by both the server and the browser

Adding Server-side PKI to Password Authentication

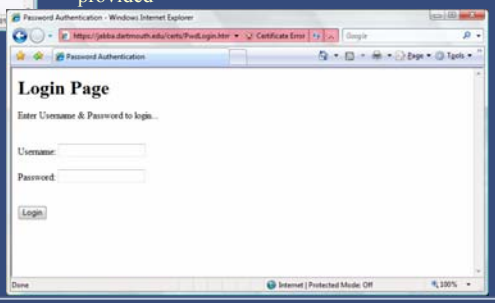


The Root CA that issued the web server certificate is trusted by both the server and the browser

Adding Server-side PKI to Password Authentication



- If another server tries to impersonate the real server, the browser warns the user about the potential deception
- If the Root CA is NOT trusted by the browser a similar warning is provided



- Even if the user ignores these warnings and continues to the web site, the browser address bar displays a red background and a broken lock to indicate the untrusted nature of the connection

Adding Server-side PKI to Password Authentication

- By using an "http" URL, the user has no guarantee that they are talking to the correct server
- There are no transport protections so username and password can be intercepted and stolen in transit (wired or wireless)
- A MITM attacker simply pretends to be the server (local DNS poisoning) , asks the user for their credentials and replays the answers to the real server in real time to gain access – the user is oblivious to this attack
- If the password is saved in the browser, it can be stolen by malware or a malicious user
- A user can be socially engineered to reveal username /password to an attacker

Risk eliminated

Risk reduced

- Passwords only provide a single authentication factor
- Passwords generally represent a poor binding between identity and credential

- The server knows everyone's password – it is a single point of failure for ALL credentials if it is compromised
- A malicious server can masquerade as any user to any other service that accepts the same username/ password (MITM)
- The server has very minimal assurance that they are talking to the original user – the password could have been shared, stolen or guessed

Adding Server-side PKI to Password Authentication

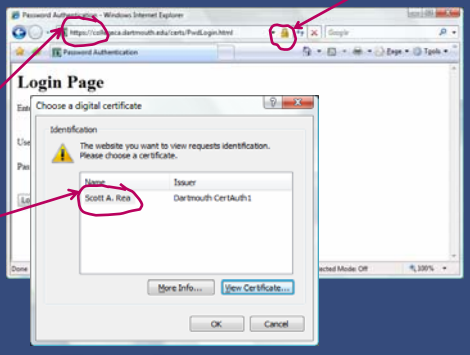
- Server-side PKI can strengthen password authentication by adding a SSL/TLS certificate to the web server and requiring users to connect over HTTPS
- Server-side certificates with TLS provides a guarantee to users that they are talking to an authentic web service – providing they trust the Root CA that issued the server certificate
- Server-side certificates with TLS provides secure communications between the browser and the server once the encrypted session has been established
- A "Man-in-the-middle" (MITM) attack is still feasible, but requires a far more complex set up to get the user to ignore the browser warnings about untrusted certificates
- The only server that can act maliciously is the original hosted service because it legitimately knows all the users credentials

- Server-side PKI only provides strong authentication of server to browser – the reverse direction authentication is still reliant on the authentication protocol being used (in this case, passwords – which are quite weak) which can lower the assurance of any transaction performed using this process

Software Certificate Authentication

Using Software Certificates with Server-side PKI for Authentication

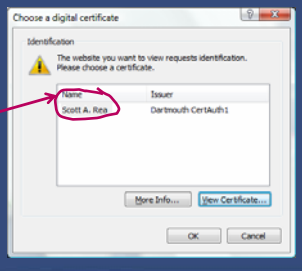
- Client-side PKI can replace password authentication when users have a digital certificate signed by a trusted Root CA and a corresponding private key
- Software certificates stored in browsers are the simplest implementation of client-side PKI
- Client-side PKI operates in conjunction with server-side PKI to provide a mutually strongly authenticated session



- Instead of providing a username/password pair, the user is asked to select a software certificate registered with the browser, as their means of authenticating

Using Software Certificates with Server-side PKI for Authentication

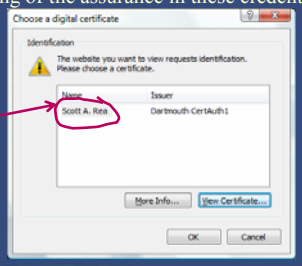
- Which client certificates are acceptable, is controlled on the server
- The server provides a list to the browser of all the Root CAs that it trusts to issue client certificates, that it will in return accept for authentication purposes
- The browser then prompts the user with a list of only those client certificates that they have (known to the browser) that the server will accept (based on the commonly trusted Root CAs between browser and server)
- Once the user chooses a certificate, the browser asks permission to use the private key associated with that certificate, to encrypt a nonce provided by the server
- The encrypted nonce along with the user's certificate is returned to the server
- If the user does not have any acceptable client certificates, they are unable to establish a secure connection



- The server uses the certificate to decrypt the nonce and verify that the user controls the private key
- Since the certificate contains only public information, and the nonce is a one-time-use only, the user is able to authenticate without revealing their authentication token (the private key) – but still providing proof of its possession

Using Software Certificates with Server-side PKI for Authentication

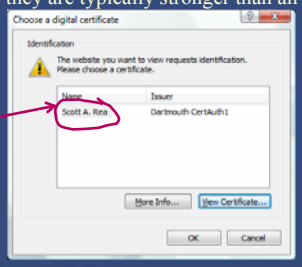
- Permission to use the private key associated with a certificate should be enabled by a password or PIN – but this is not required, and can only be enforced via policy and not technologically with current modern browsers
- If a private key stored in the browser is protected by a password, then this technically provides two factors of authentication – something the user has (a private key), and something the user knows (a password)
- Some browsers allow back-up and export of private keys, meaning there could be many locations that a given private key exists. There is no way to know what protections are being used for private keys being stored in this way. This situation can potentially lead to a lowering of the assurance in these credentials
- Restrictions on private back-up and storage can be controlled via policy, but do not negate malicious users



- Users who are unable to provide assurances regarding ALL copies of their private key, effectively reduce the authentication strength via this method to something closer to a single factor of authentication – since they can no longer guarantee that the private key is something ONLY they have

Using Software Certificates with Server-side PKI for Authentication

- Client certificates that do not have their corresponding private keys protected by passwords (or whose passwords are remembered in the browser), may be compromised by malware, used maliciously on the client's browser or copied to another location
- Issuance of client certificates is governed by policies created to ensure a strong binding between the authentication token (the private key) and the identity of its owner, and also govern how a user should protect their private key through its life-cycle. These controls are typically much stronger than those for passwords
- Client certificates stored in software are not as portable as passwords, but provide much stronger authentication – they are typically stronger than an equivalent 128 character password



- Users who are unable to provide assurances regarding ALL copies of their private key, effectively reduce the authentication strength via this method to something closer to a single factor of authentication – since they can no longer guarantee that the private key is something ONLY they have

Using Software Certificates with Server-side PKI for Authentication

- By using an "http" URL, the user has no guarantee that they are talking to the correct server
- There are no transport protections so username and password can be intercepted and stolen in transit (wired or wireless)
- A MITM attacker simply pretends to be the server (local DNS poisoning) , asks the user for their credentials and replays the answers to the real server in real time to gain access – the user is oblivious to this attack
- If the password is saved in the browser, the private key may be stolen by malware or a malicious user
- A user can be socially engineered to reveal username/password to an attacker

- Passwords only provide a single authentication factor
- Passwords generally represent a poor binding between identity and credential
- The server knows everyone's password – it is a single point of failure for ALL credentials if it is compromised
- A malicious server can masquerade as any user to any other service that accepts the same username/password (MITM)
- The server has very minimal assurance that they are talking to the original user – the password could have been shared, stolen or guessed

Risk eliminated

Very Low Risk

Risk reduced

Using Software Certificates with Server-side PKI for Authentication

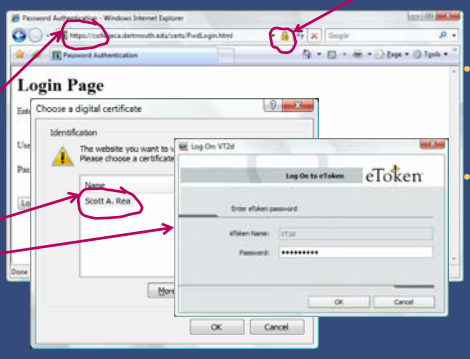
- Software certificates replacing passwords as the means of client authentication, when combined with server-side PKI, can eliminate almost all the risks inherent in password based protocols
- Potential for poor protection of private keys through inappropriate (or no) passwords can reduced the effective level of assurance on this type of authentication, such that it cannot be considered true two factor authentication
- Social engineering attacks must be much more complex to steal the user's password and then subsequently gain access to the user's private key
- A server has a reasonable assurance that it is communicating with the original authenticated user
- MITM attacks while still possible are highly unlikely due to the additional factors that must be cater for

- Software certificates as a means of authentication are far superior to plain passwords in every aspect – however there is one deficiency that they introduce – they are not very portable between multiple legitimate systems
- At Dartmouth we mitigate this issue by making it easy to get additional certificates on subsequent or additional legitimate systems

Virtual Tokens Authentication

Using PKI Virtual Tokens with Server-side PKI for Authentication

- PKI Virtual Tokens work exactly the same as software certificates from a functionality perspective. The user interactions are exactly the same, except when the browser asks permission to use the private key associated with that certificate, the user MUST provide the password to the Virtual Token.
- The advantage of private keys stored in Virtual Tokens and registered with the browser is that strong local password protection of private keys can be enforced.
- A Virtual Token is really nothing more than a secure software container for the private key instead of using the browser's key store



- Virtual Tokens also remedy the portability deficiencies introduced by software certificates
- Virtual Tokens still suffer from the issue of being able to duplicate multiple instances of the private key such that the assurance factors of this method is reduced

Using PKI Virtual Tokens with Server-side PKI for Authentication

- By using an "http" URL, the user has no guarantee that they are talking to the correct server
- There are no transport protections so username and password can be intercepted and stolen in transit (wired or wireless)
- A MITM attacker simply pretends to be the server (local DNS poisoning) , asks the user for their credentials and replays the answers to the real server in real time to gain access – the user is oblivious to this attack
- If the password is saved in the browser, the private key may be stolen by malware or a malicious user
- A user can be socially engineered to reveal username/password to an attacker

Risk eliminated

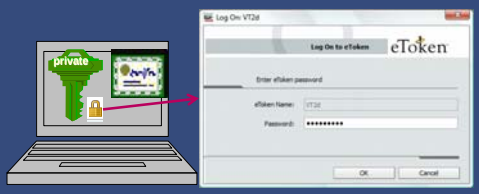
Very Low Risk

Risk reduced

- Passwords only provide a single authentication factor
- Passwords generally represent a poor binding between identity and credential
- The server knows everyone's password – it is a single point of failure for ALL credentials if it is compromised
- A malicious server can masquerade as any user to any other service that accepts the same username/password (MITM)
- The server has very minimal assurance that they are talking to the original user – the password could have been shared, stolen or guessed

Using PKI Virtual Tokens with Server-side PKI for Authentication

- PKI Virtual Tokens are a relatively new product offering from most organizations – there are still teething issues being experienced by vendors deploying this technology
- Virtual Tokens provide a lower cost of implementation than the next step up in authentication assurance level which is hardware tokens
- Virtual Tokens mitigate most (but not all) of the remaining risks that software certificates did not address and facilitate portability of credentials which was a deficiency that reliance on software certificates introduced
- Virtual Tokens cannot however, provide true two-factor authentication because they can be duplicated an infinite number of times and thus are vulnerable to offline brute force or password guessing attacks

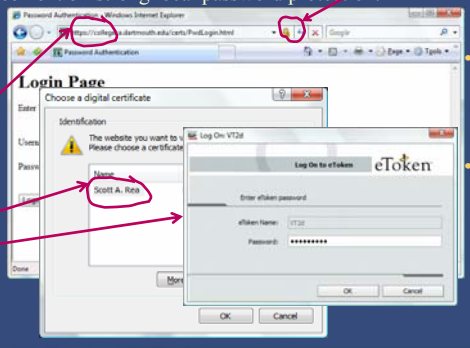


- NOTE: Virtual Tokens may also come in a one-time-pad (OTP) format. These types of tokens are more susceptible to MITM than PKI based tokens. In the continuum of authentication assurance, OPT tokens fall close to (but below) software certificates due to their password related deficiencies

Hardware Tokens Authentication

Using PKI Hardware Tokens with Server-side PKI for Authentication

- PKI Hardware Tokens work exactly the same as software certificates or Virtual Tokens from a functionality perspective. The user interactions are exactly the same, except when the browser asks permission to use the private key associated with that certificate, the user MUST provide the password to the Hardware Token after having connected the device.
- The advantage of private keys stored in Hardware Tokens and registered with the browser is the enforcement of strong local password protection.
- A Hardware Token is a secure container for the private key instead of using the browser's key store



- Hardware Tokens also remedy the portability deficiencies introduced by software certificates
- Hardware Tokens restrict the export of private keys such that two factor authentication can be achieved

Using PKI Hardware Tokens with Server-side PKI for Authentication

- By using an "http" URL, the user has no guarantee that they are talking to the correct server
- There are no transport protections so username and password can be intercepted and stolen in transit (wired or wireless)
- A MITM attacker simply pretends to be the server (local DNS poisoning) , asks the user for their credentials and replays the answers to the real server in real time to gain access – the user is oblivious to this attack
- If the password is saved in the browser, the private key may be stolen by malware or a malicious user
- A user can be socially engineered to reveal private key password to an attacker

Risk eliminated

Very Low Risk

Risk reduced

- Passwords only provide a single authentication factor
- Passwords generally represent a poor binding between identity and credential
- The server knows everyone's password – it is a single point of failure for ALL credentials if it is compromised
- A malicious server can masquerade as any user to any other service that accepts the same username/password (MITM)
- The server has very minimal assurance that they are talking to the original user – the password could have been shared, stolen or guessed

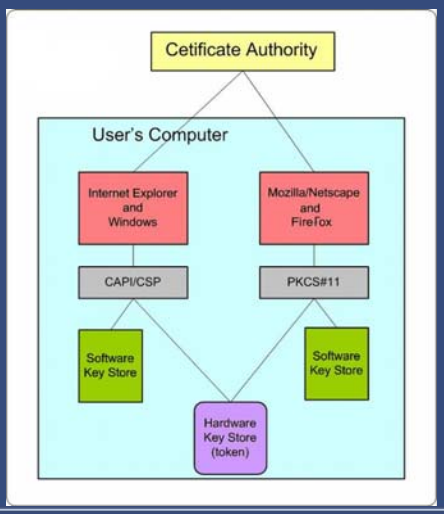
Using PKI Hardware Tokens with Server-side PKI for Authentication

- PKI Hardware Tokens are a mature technology that is well regulated – the FIPS 140 Standard (about to release version3) provides ratings for cryptographic devices of which Hardware Tokens are a subset
- Hardware Tokens require a higher initial cost of implementation than other options discussed here, but this may be offset by the cost of higher helpdesk support for other options e.g. password resets, revocation of certificates due to lost files or compromised passwords
- Hardware Tokens mitigate or solve all risks that software certificates did not address and yet facilitate portability of credentials which was a deficiency that reliance on software certificates introduced
- Hardware Tokens provide true two-factor authentication for the highest level of assurance in authentications of any method discussed

- NOTE: PCI and HIPAA both require two factor authentication under certain circumstances. Hardware Tokens are the only authentication technology currently in operation at Dartmouth that can provide this

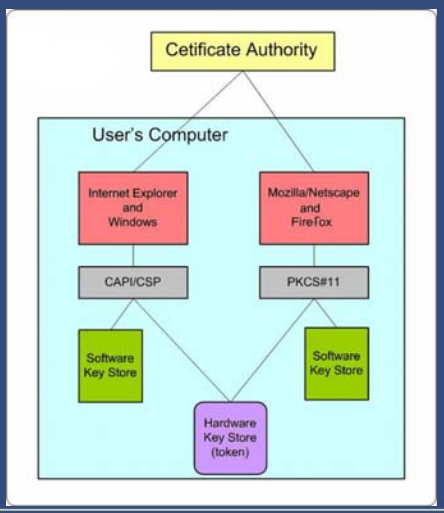
Hardware Tokens provide Portability for PKI Private Keys

- Hardware based private key stores provide cross-browser and cross-platform interoperability
  - PCMCIA, Smartcard or USB form factors available
  - USB was Dartmouth's choice because it is relatively ubiquitous and does NOT require the installation of additional hardware i.e. card readers
  - Aladdin eTokens were chosen because they best supported the range of platforms Dartmouth wished to support (i.e. Windows, Mac, Linux)



eTokens provide the Strongest Protection of PKI Private Keys

- Hardware private key stores provide enforcement of private key security
  - Hardware devices can ensure that only one copy of the key exists – it is generated in hardware and cannot be exported (never leaves the token)
  - Hardware devices can ensure that applications do not have direct access to private keys
  - Hardware devices can ensure that passwords are set to protect private keys, and can limit attempts to guess them
  - Because of portability functionality of hardware devices (detailed previously), there is no need for users to manage import and export between locations
  - Aladdin eTokens were chosen because they are FIPS 140 level 3 certified hardware devices



Summary

Identity Assurance is a measure of the confidence that the entity at the other end of an authentication event, is who they are claiming to be

- Identity Assurance is a pre-requisite for effective identity management, and identity management a pre-requisite for robust security
- The strength of any Authentication event is dependent on the following:
  - The original process to bind the identity to the authentication token
  - The life cycle management and protection of the authentication token by the identity
  - The infrastructure and protocols used by a service to validate an authentication token
  - The use of multiple authentication factors to verify identity

Summary

- A flexible identity management infrastructure should support different levels of authentication – including:
  - High assurance credentials on hardware tokens
  - Medium assurance credentials on virtual tokens
  - Low assurance credentials with PKI software certificates
  - Rudimentary assurance password based credentials
- It is recommended that institutions of higher education move away from password based authentication as soon as possible (where practical) for any type of sensitive data access

While debate continues on what type of technology is best suited to prevent identity theft, many experts believe that a combination of PKI infrastructure and two-factor authentication offers the greatest promise of protection.

Source: Financial Services Technology, Preventing Identity Theft

Questions?

For more information…

Scott Rea - Scott.Rea@dartmouth.edu